Breaking free from the chains of blockchain protocols

Yonatan Sompolinsky

joint work: Aviv Zohar, Yoad Lewenberg

The Hebrew University, Jerusalem
Israel

Our daydream for Bitcoin

long-term vision.

several orders of magnitude increase in transaction volume (many MB per sec)

transactions confirmed quickly (~1 sec)

highly secure

This work

a new family of blockchain protocols

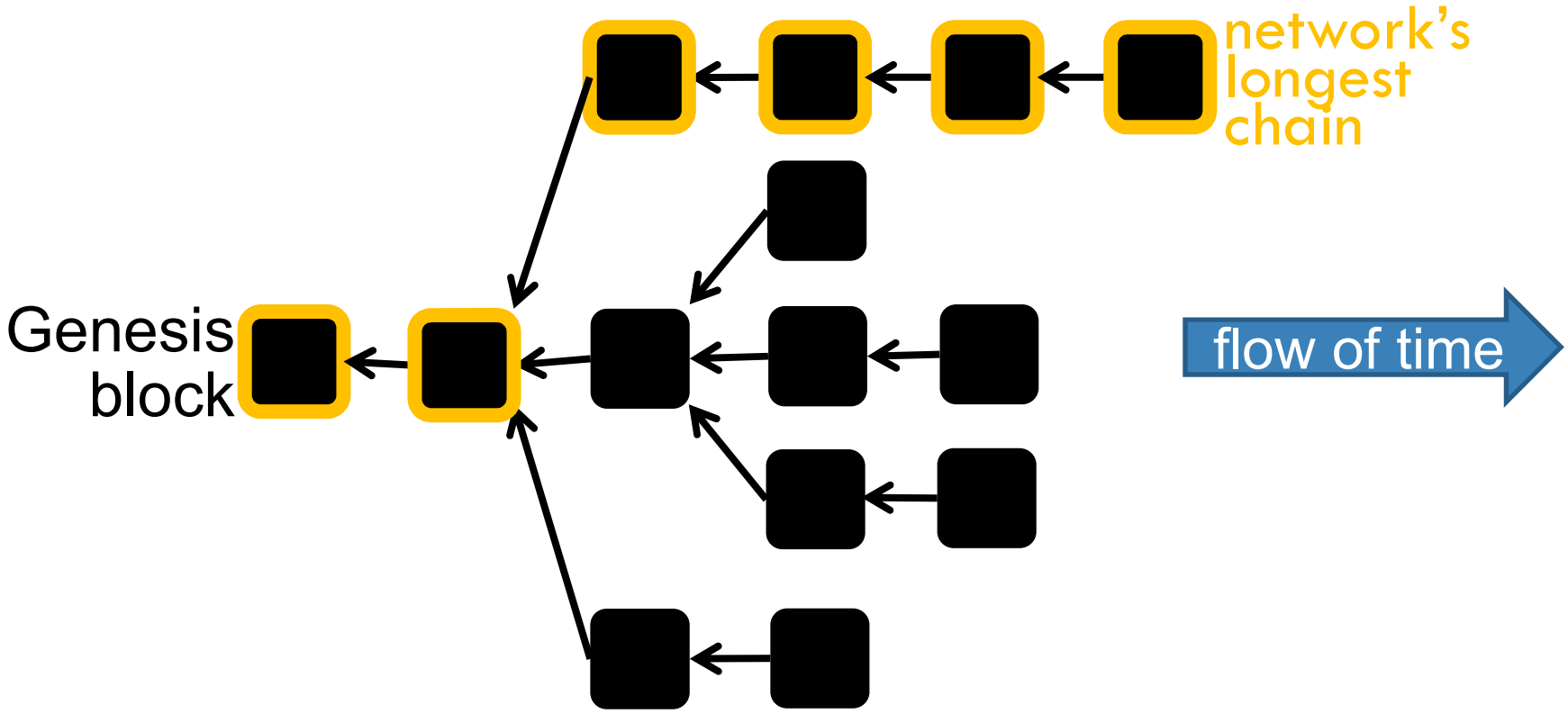
makes use of ordinary Bitcoin blocks

- orthogonal to “offchain” solutions

scalable

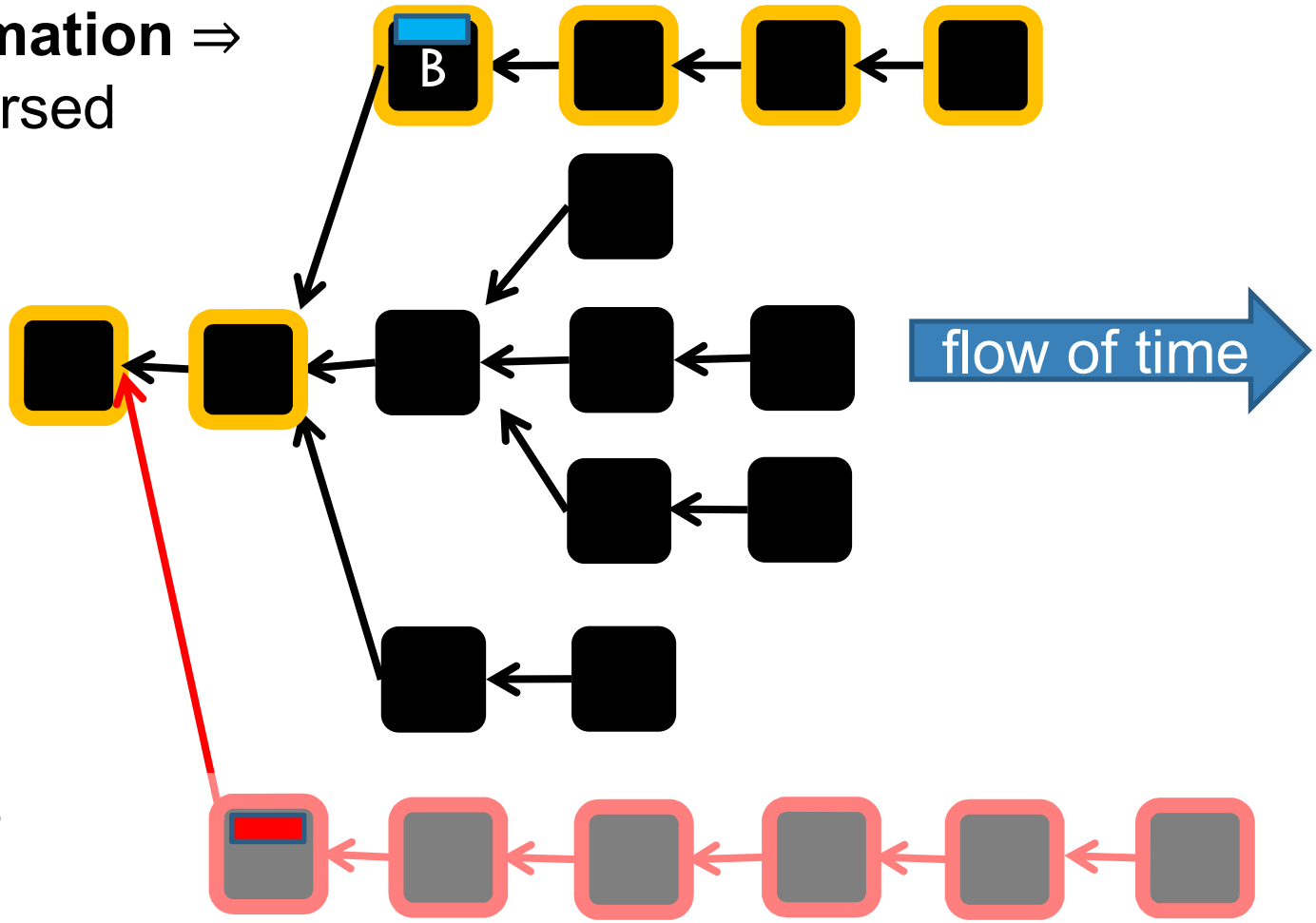
Scalingbitcoin

Public ledger under high throughput

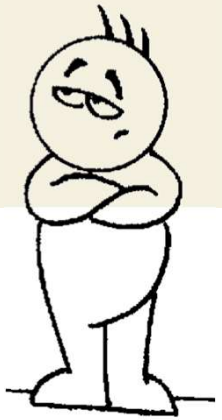


Double-spending

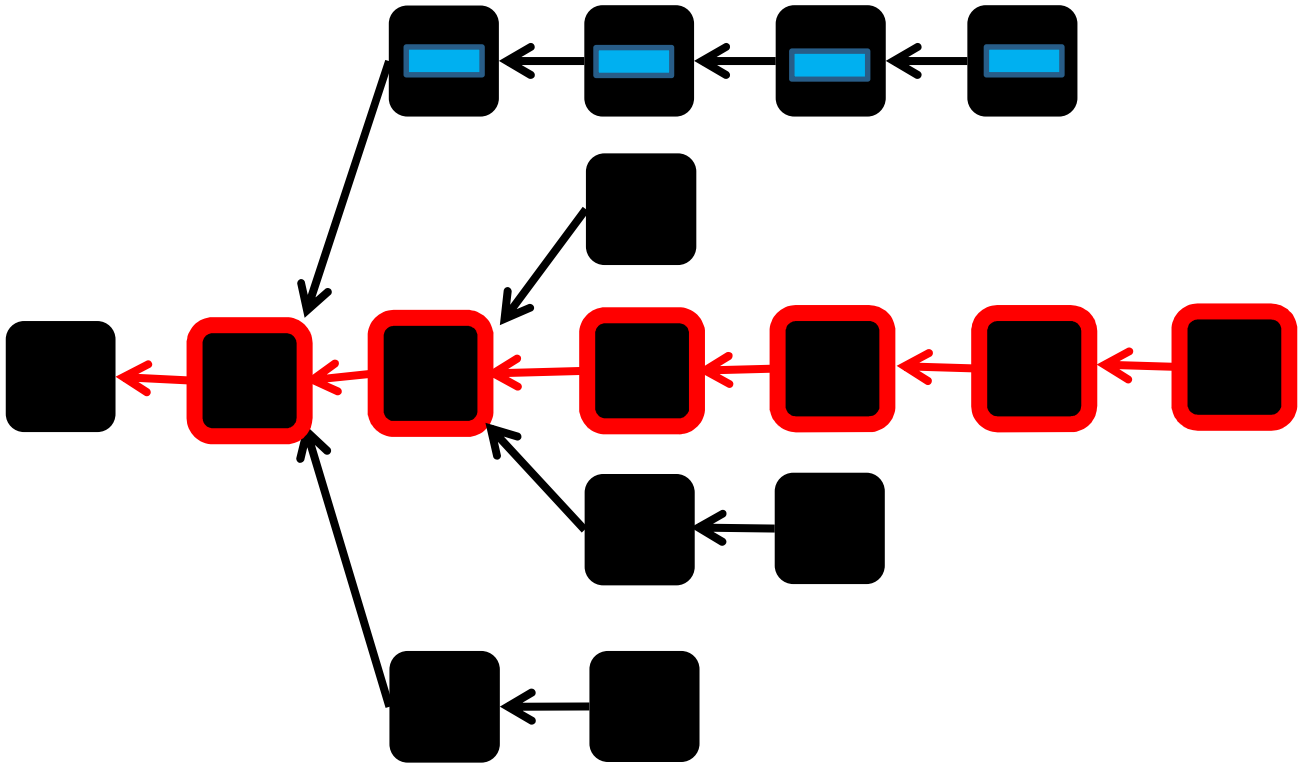
attacker publishes blocks
after confirmation ⇒
tx reversed



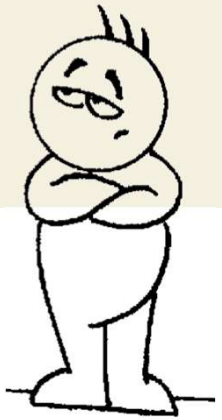
Censorship



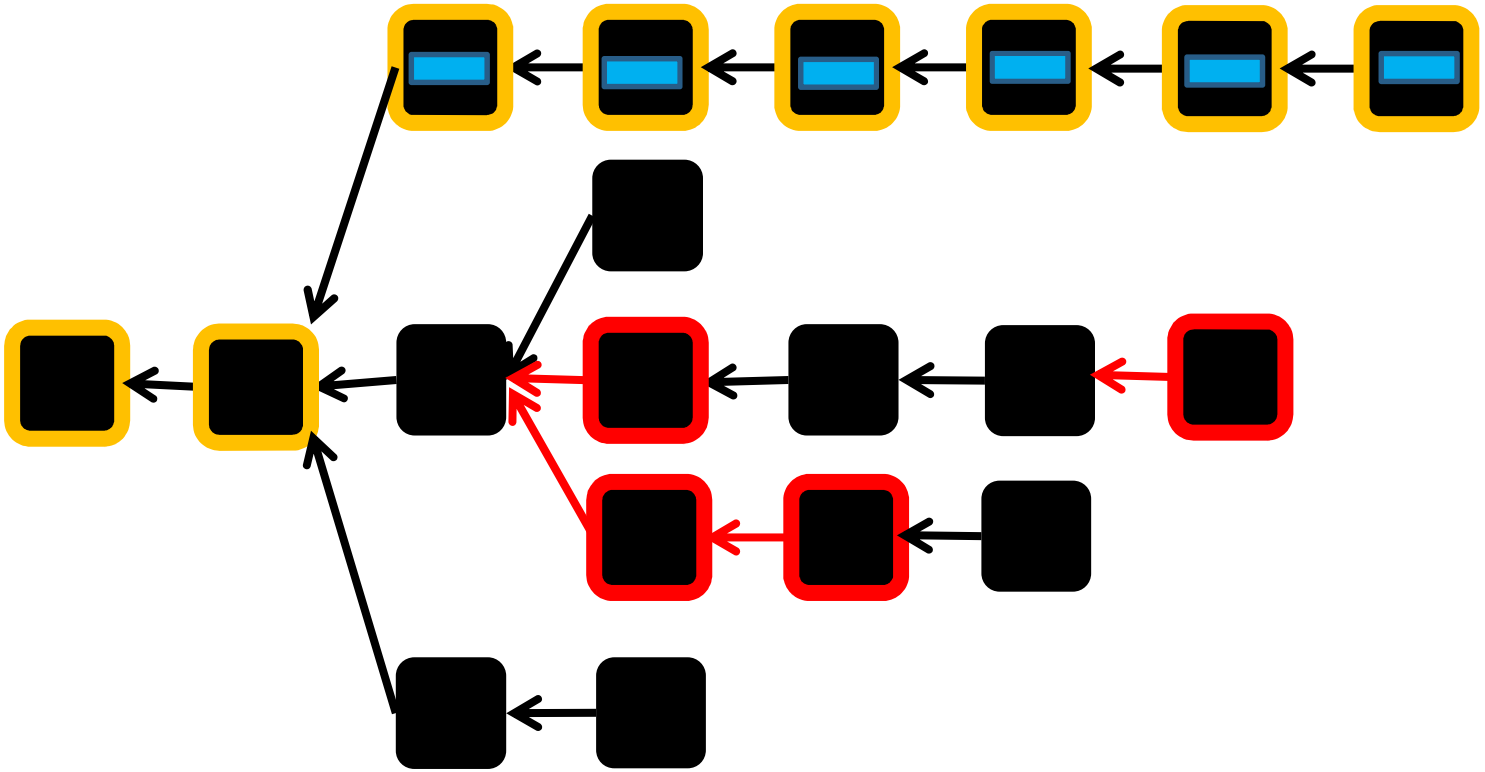
to prevent confirmation,
attacker publishes empty blocks fast



Delayed-Acceptance



to prevent confirmation,
attacker helps weaker chains to survive



Security thresholds under high throughput

	longest-chain	GHOST	ideal
double-spending	≪50%	50%	50%
censorship	≪50%	≪50%	50%
delayed-acceptance	50%	≪50%	50%

does an ideal protocol exist?

INSIGHT

"chainless" protocols.

1. to avoid vulnerability, all blocks participate in tx confirmation.
2. agreeing on the order of incoming transactions is enough.

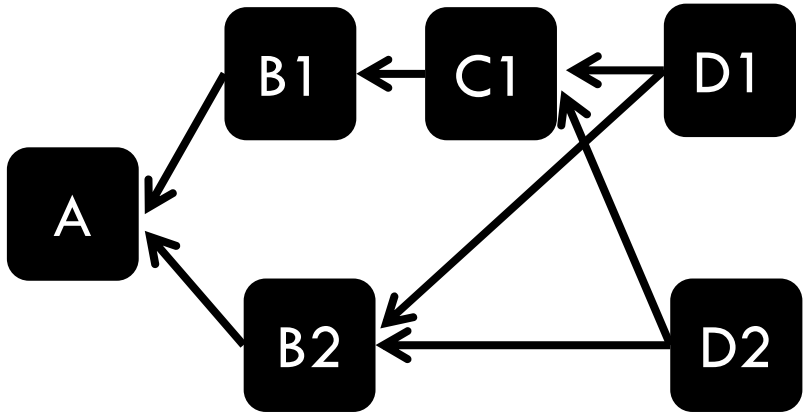
From chains to DAG

acknowledge **all predecessors**

ledger becomes block ~~chain~~ **DAG**

need to define new consistency rules

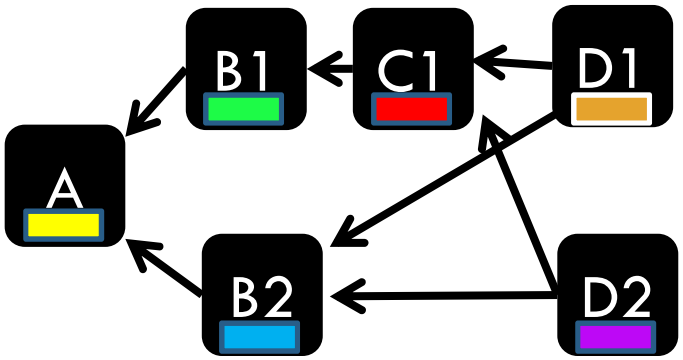
more complex, more powerful



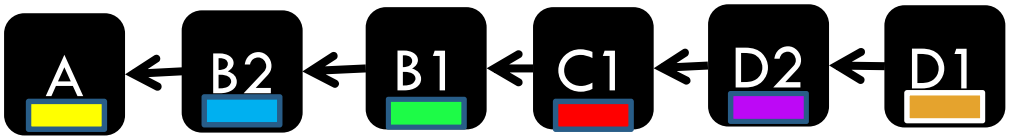
Consistency via linear ordering

observation: a **linear ordering** of blocks in DAG

induces natural consistency



block DAG



linearized block DAG

Consistency via linear ordering

a “good” order should be resilient to revision.

with high probability:

$(b \prec c)$ now implies $(b \prec c)$ later

no cutting in line: b published earlier than c implies $(b \prec c)$

not easy to satisfy - naïve protocols are manipulable

New protocol

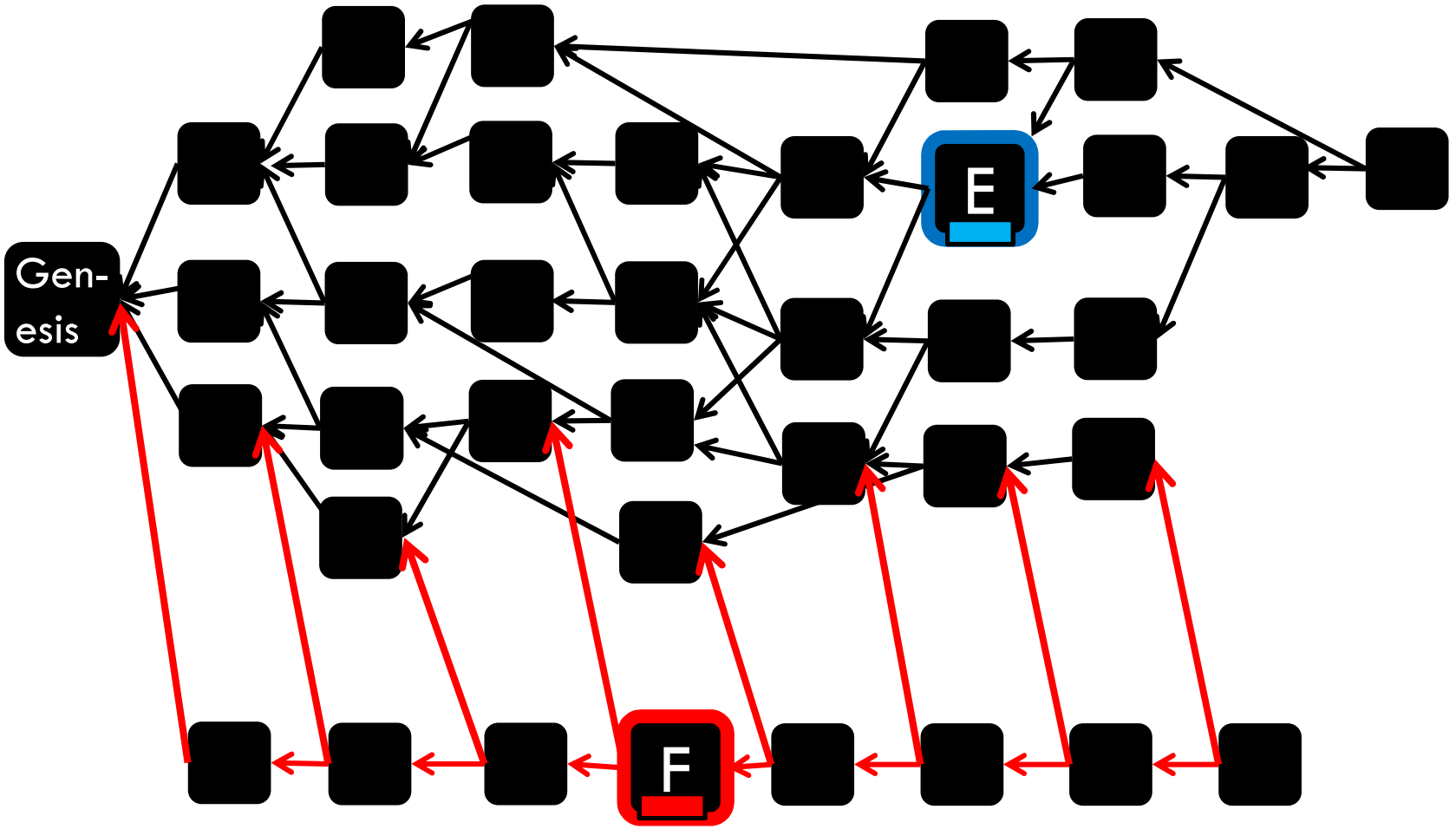
the order on a given DAG is decided by all blocks voting.

1. for each pair E, F, vote between $E \prec F$ and $F \prec E$
   a) block C that knows E or F (or both): compute vote recursively (base case: C votes $C \prec E$ for all E that it does not know)
   b) block C that knows neither E nor F: compute vote by majority of blocks in C's future
2. linearize pairwise votes via the "Schulze Method"

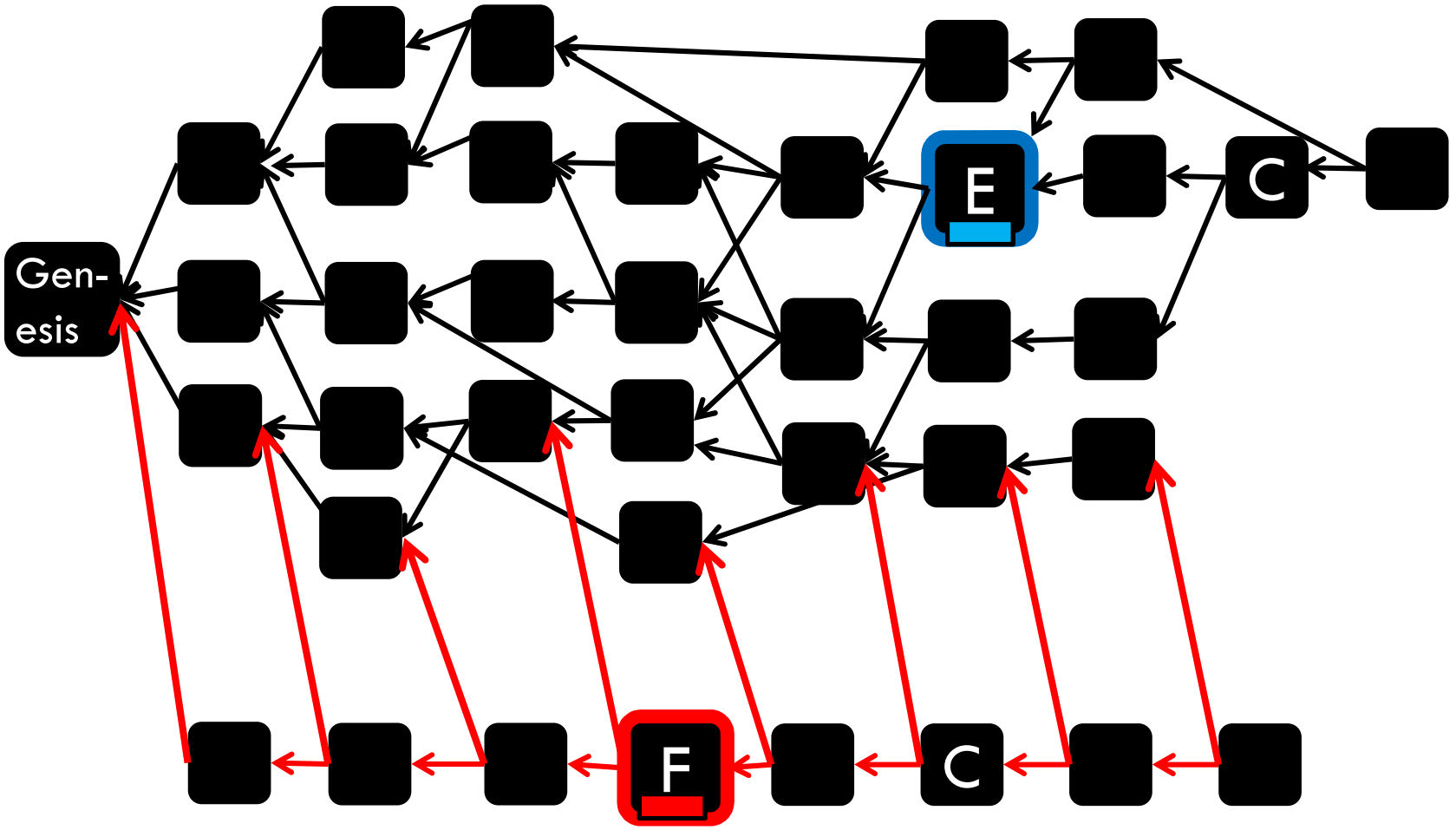
under no delays coincides with longest-chain/GHOST

provably correct

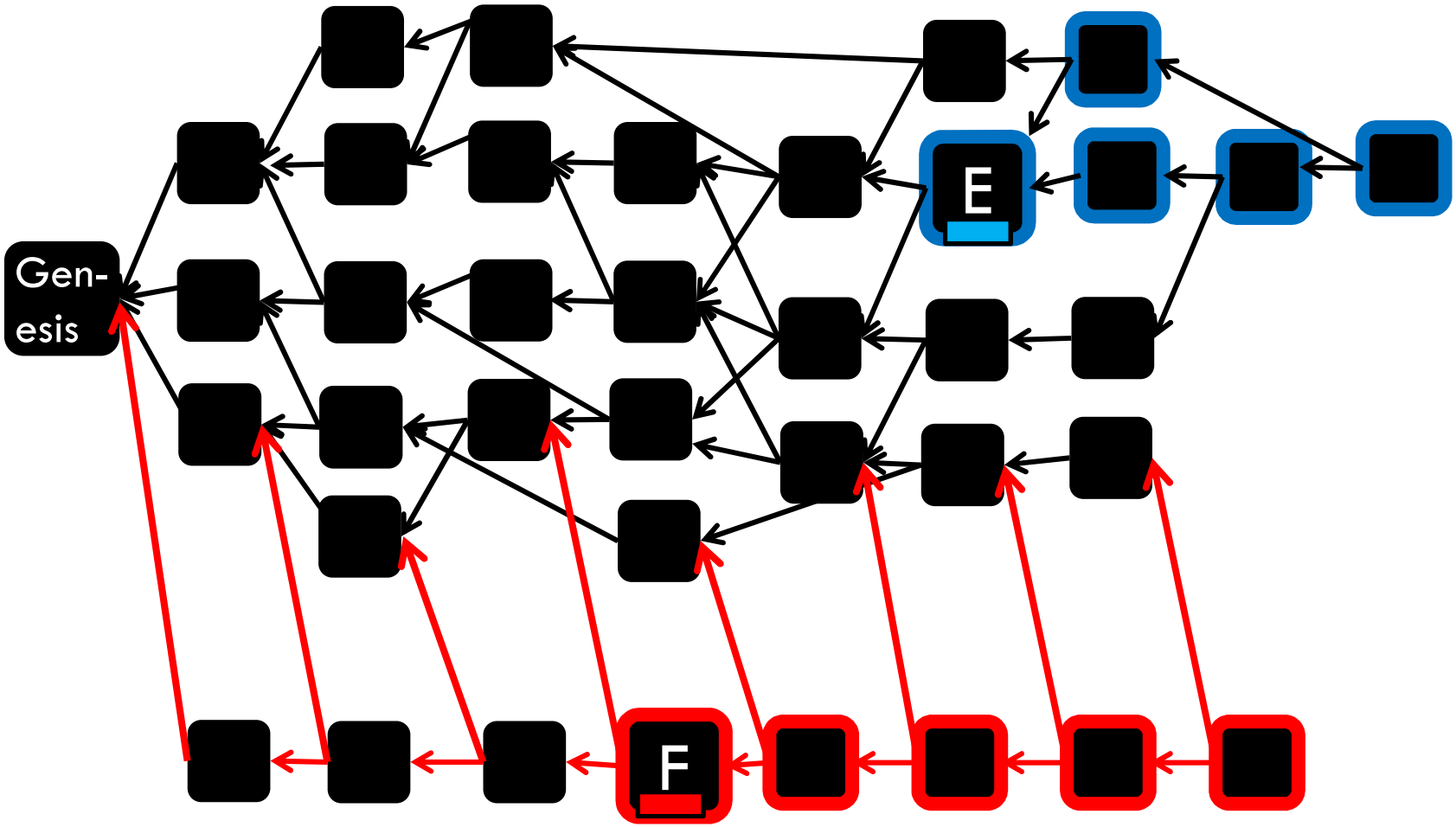
every new block runs protocol on its world-view



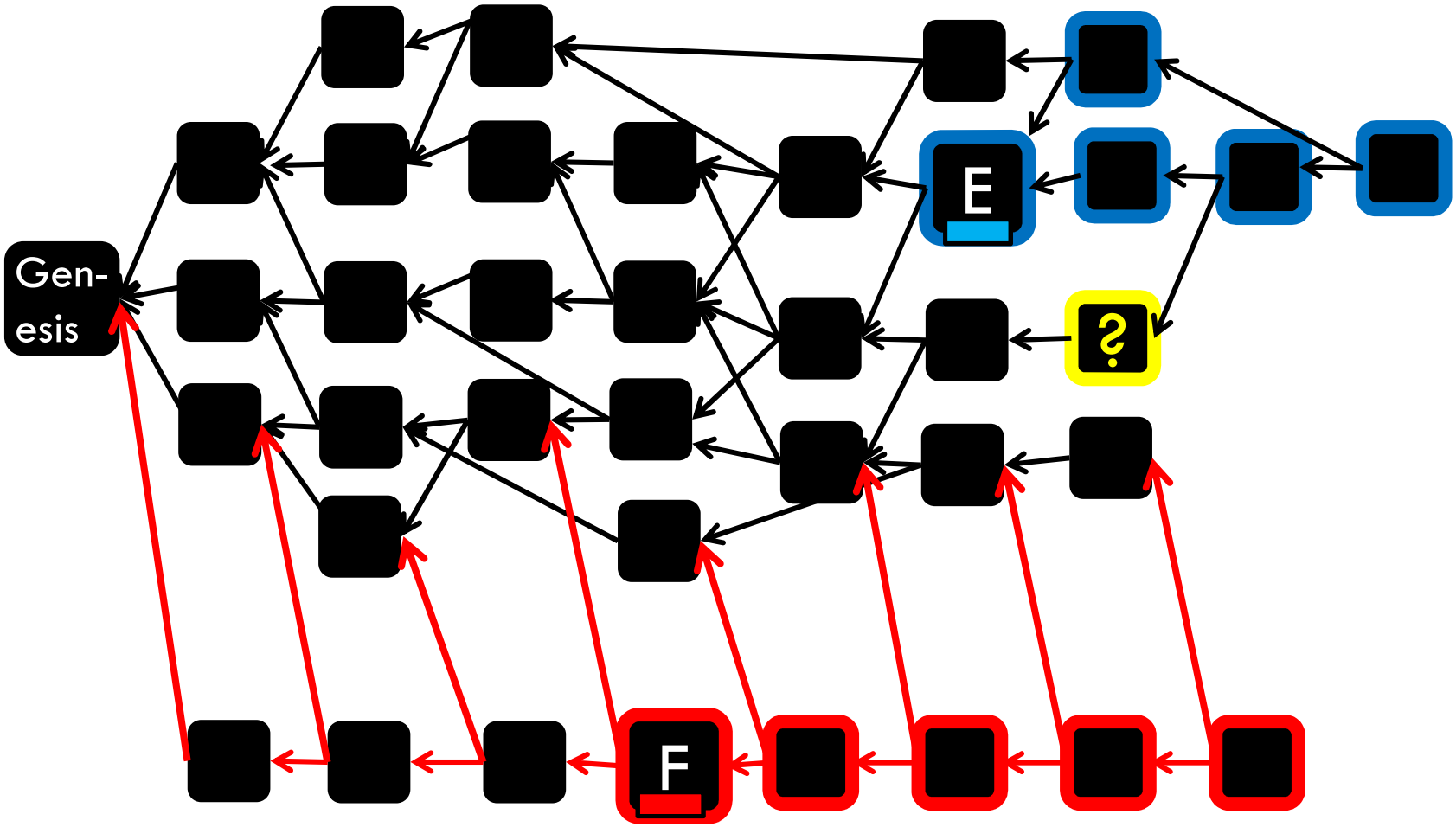
case #1: E or F are in past(C) ⟹ infer C's vote via recursion



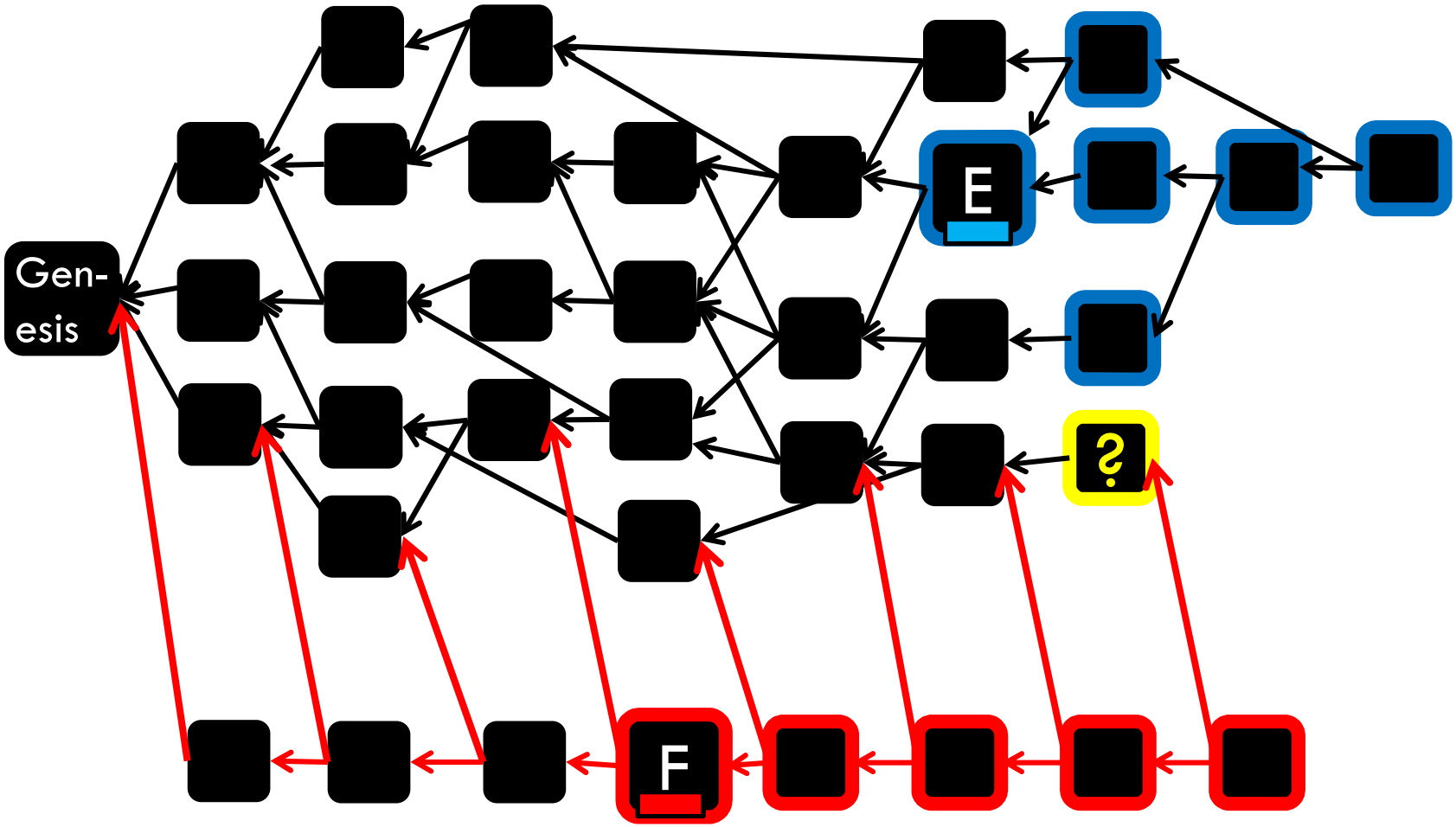
case #1: E or F are in past(C) ⟹ infer C's vote via recursion



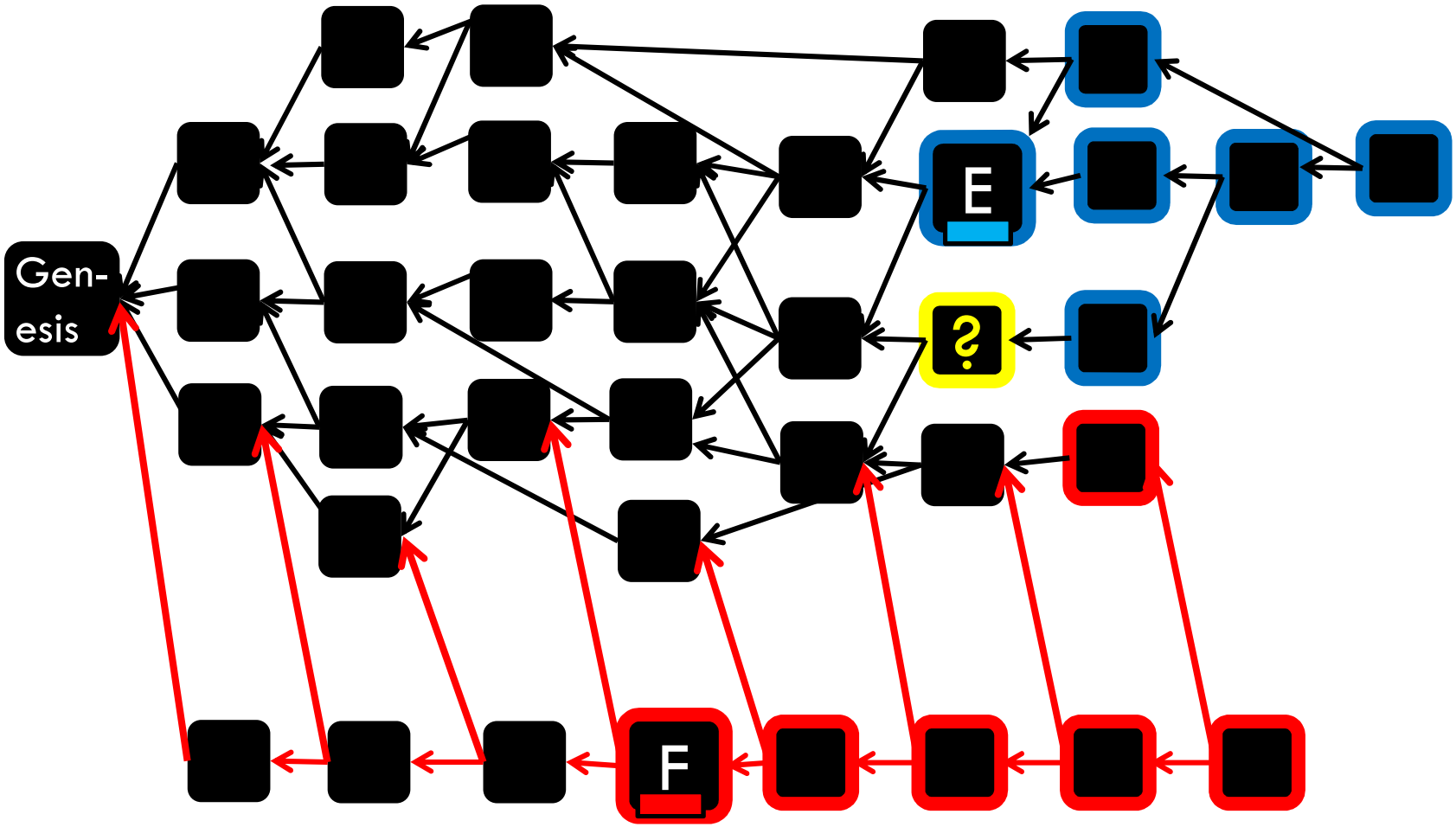
case #2, E and F not in past(C) ⇒ C votes according to majority in future (C)



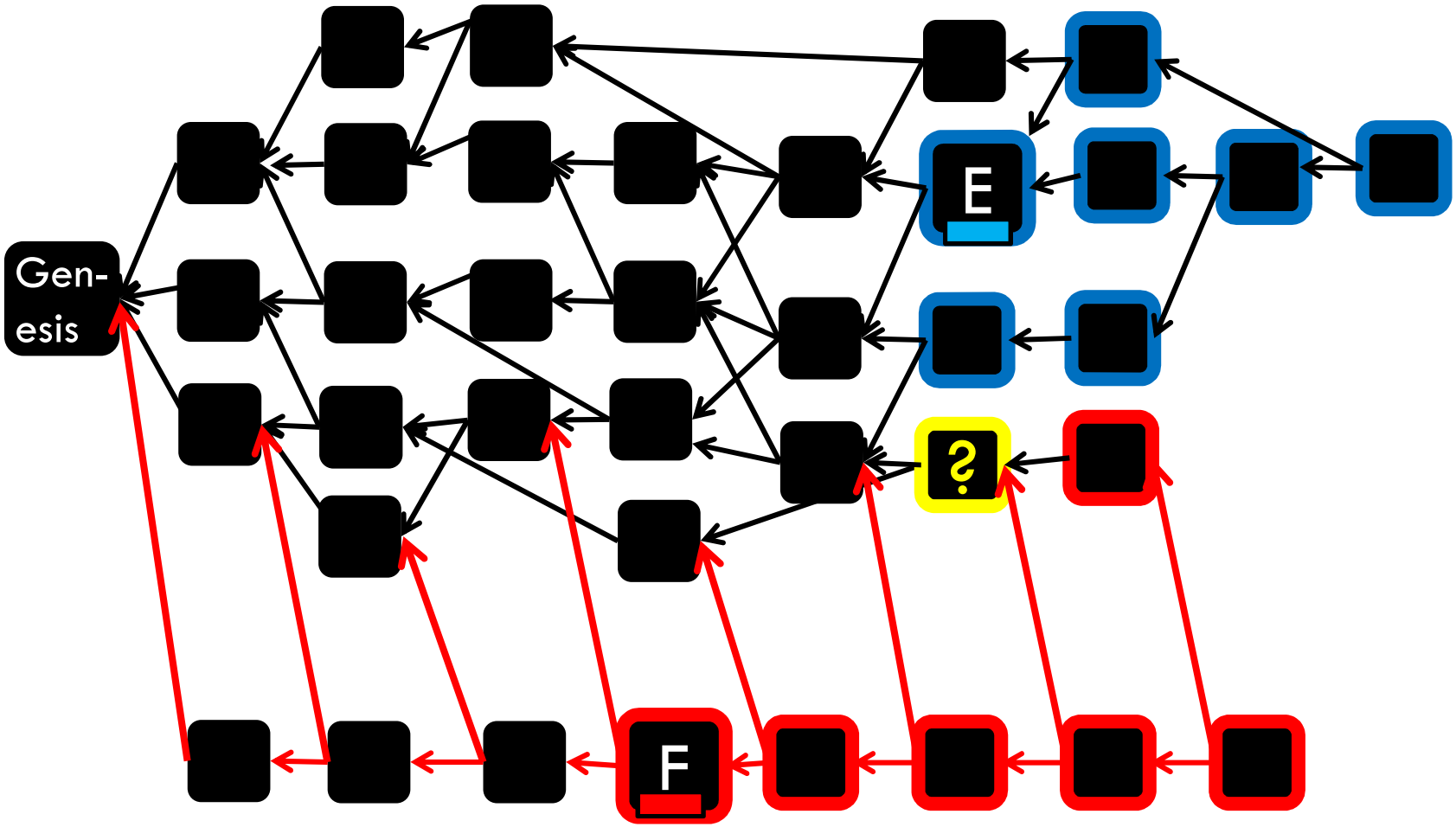
case #2, E and F not in past(C) ⇒ C votes according to majority in future (C)



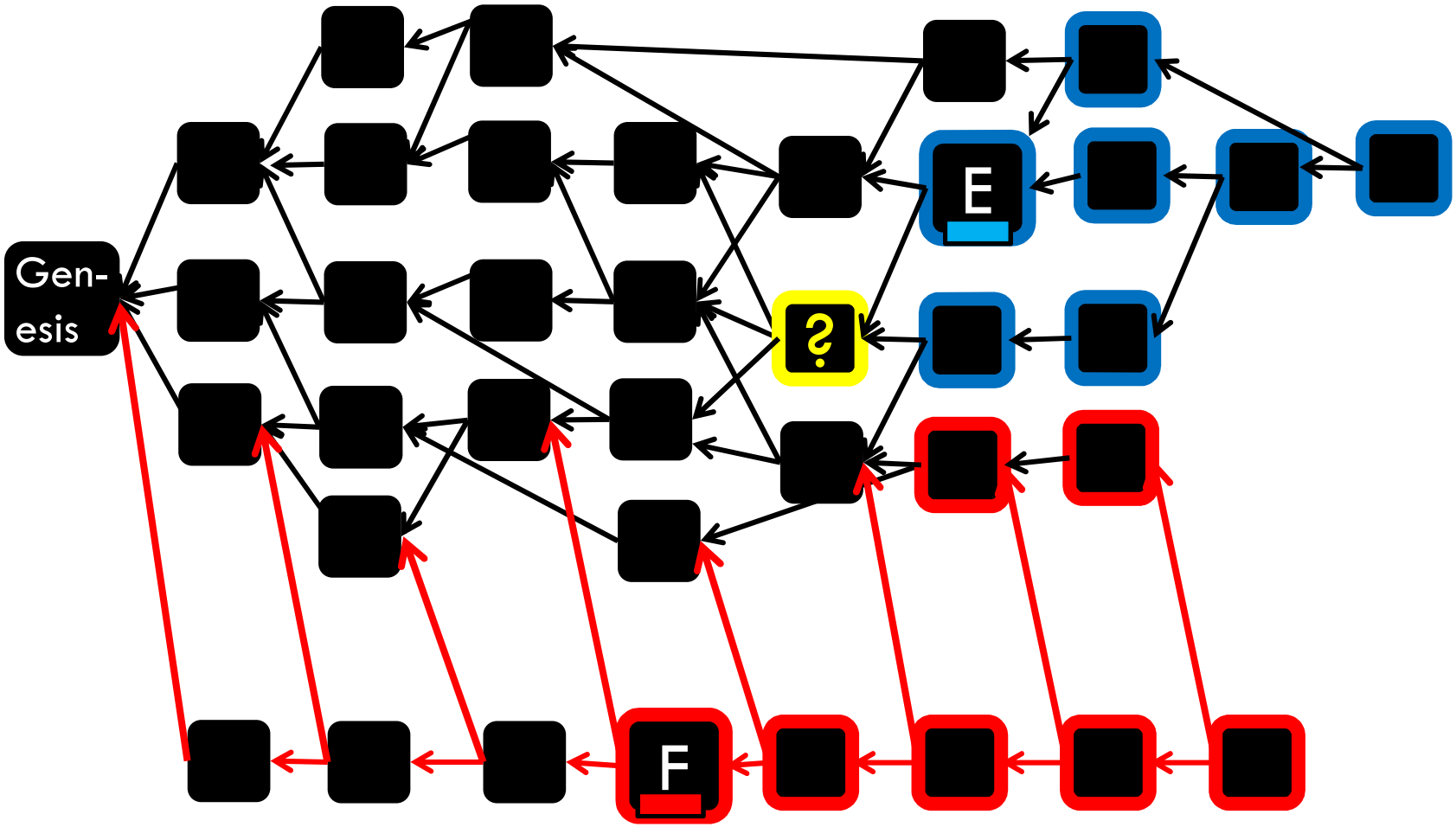
case #2, E and F not in past(C) ⇒ C votes according to majority in future (C)



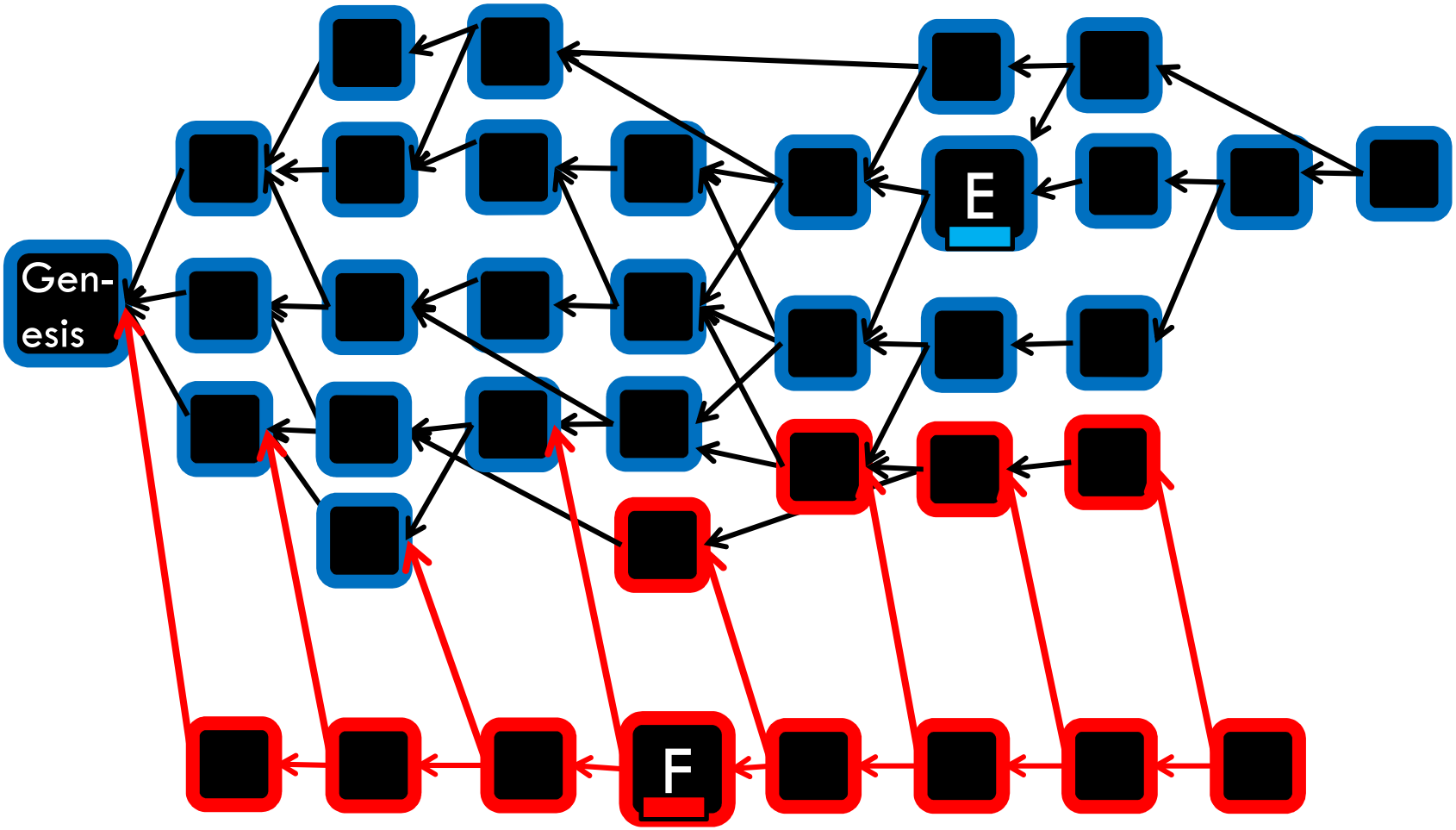
case #2, E and F not in past(C) ⇒ C votes according to majority in future (C)



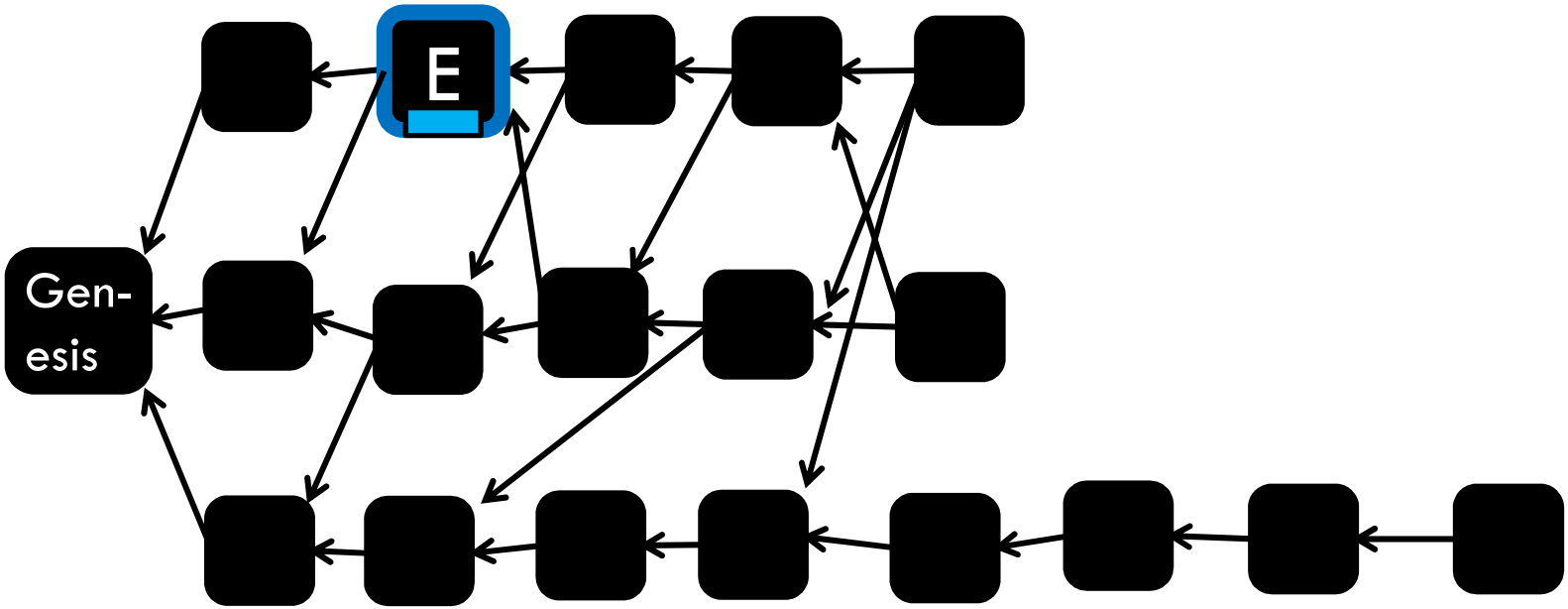
case #2, E and F not in past(C) ⇒ C votes according to majority in future (C)



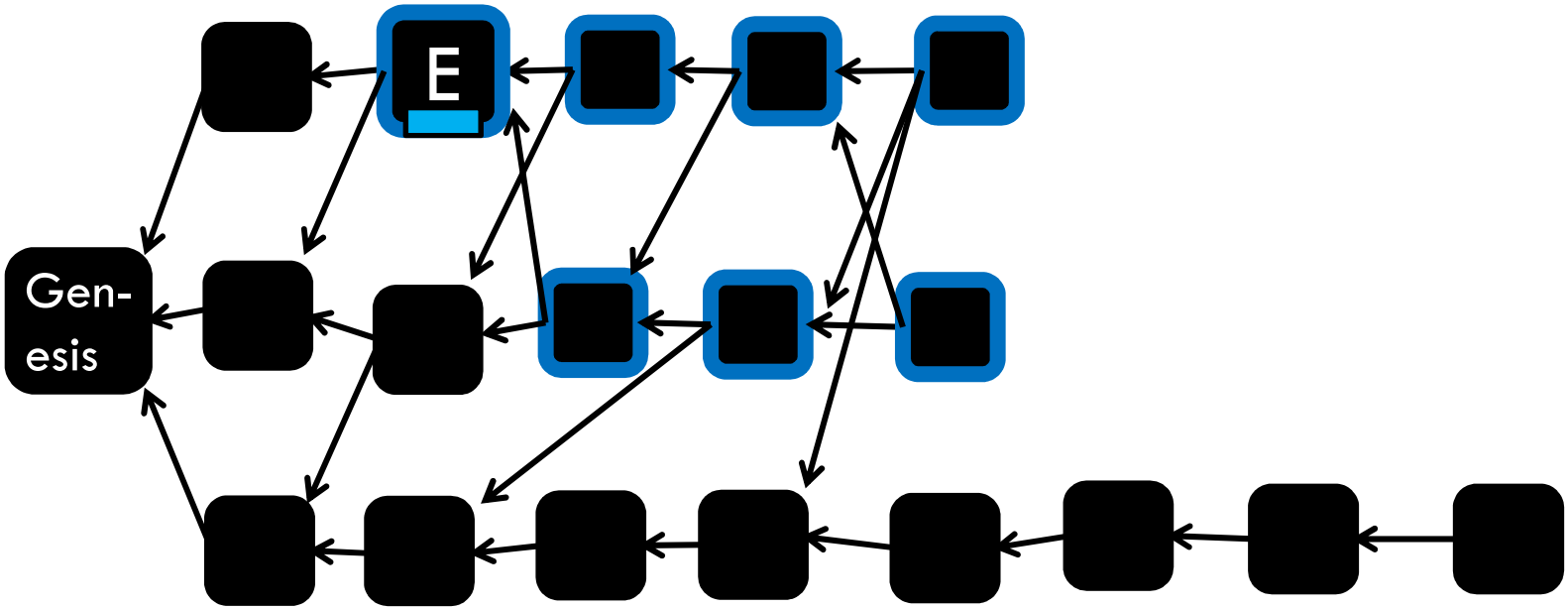
case #2, E and F not in past(C) ⇒ C votes according to majority in future (C)



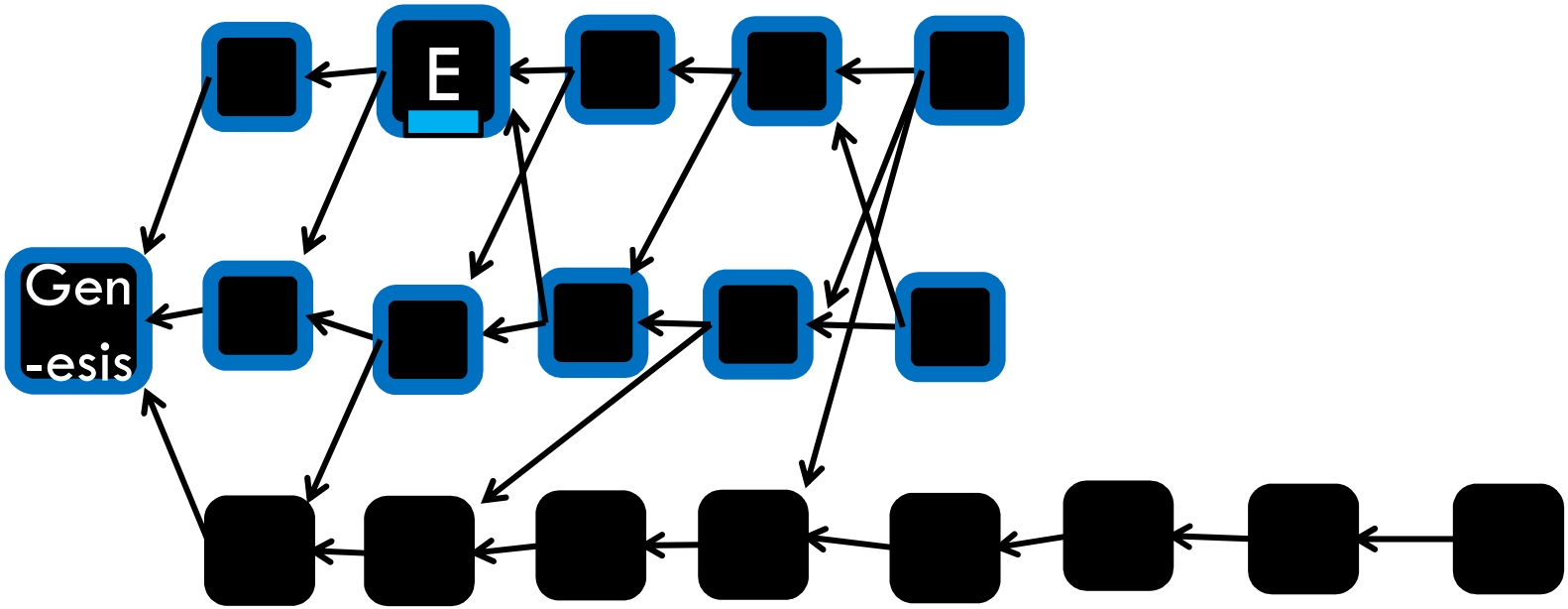
How censorship fails



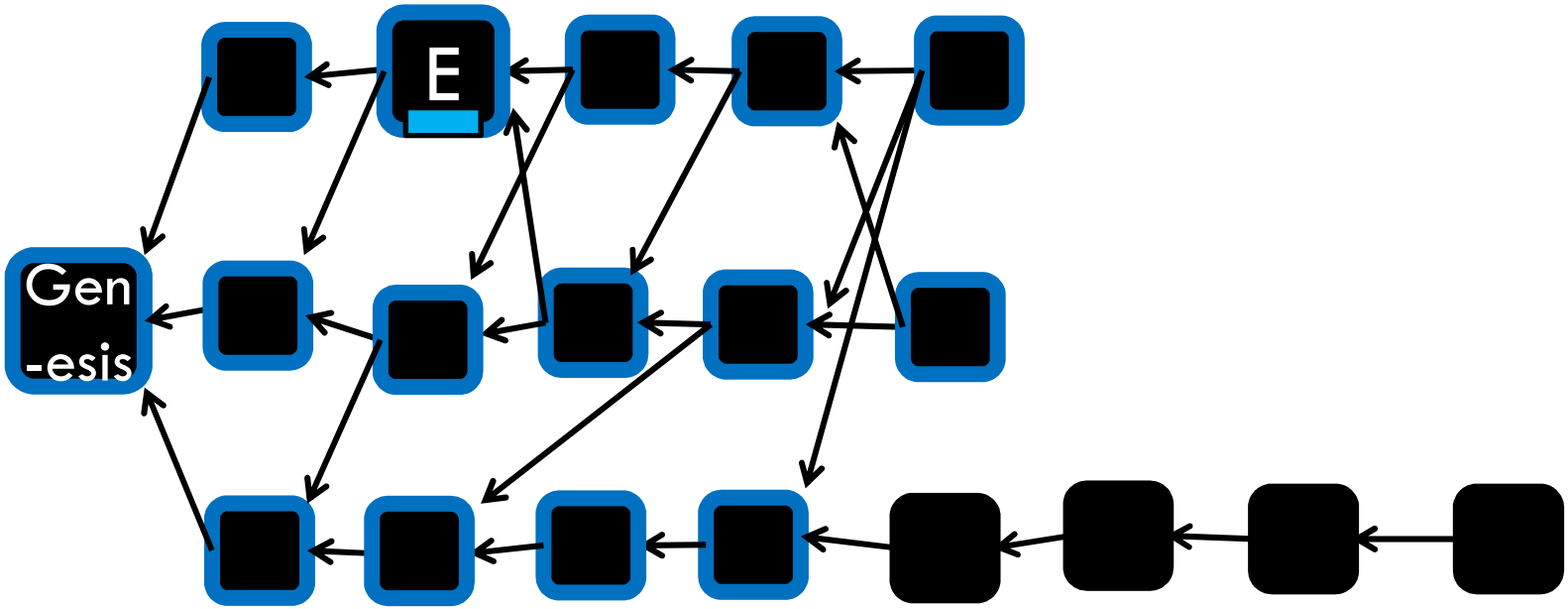
How censorship fails



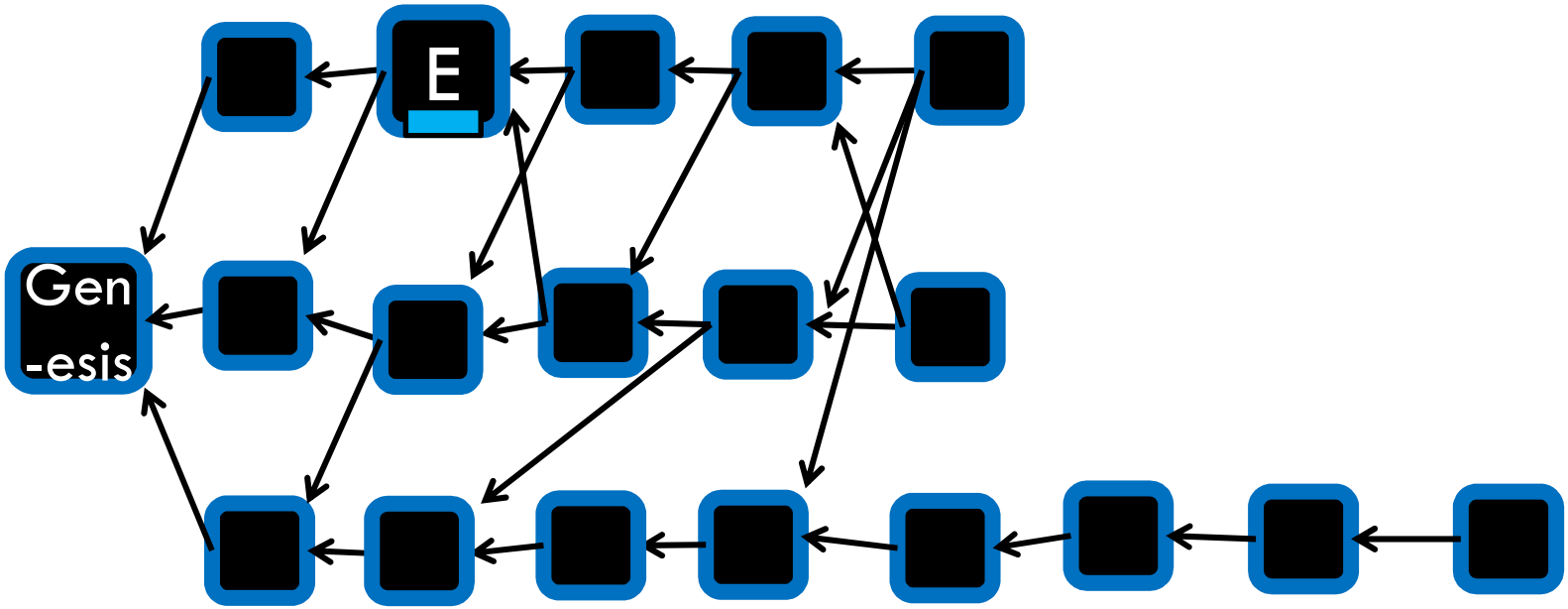
How censorship fails



How censorship fails



How censorship fails



Linearization

pairwise ordering not enough - cycles may form:

A>B>C>A

we use the “Schulze Method” to finalize the linear ordering

counting confirmations - we provide a policy,

which is faster than longest-chain/GHOST

Back to the security thresholds

	Longest-chain	GHOST	New Protocol
Double-spending	**≪50%**	**50%**	**50%**
Censorship	**≪50%**	**≪50%**	**50%**
Delayed-acceptance	**50%**	**≪50%**	**50%***

* excluding a *visible* double-spending

we prove

We are working on…

patch for mining fees - countering “delayed-acceptance” attack

considering incentives, selfish mining

- our protocol does not make things worse

SPV and compact proofs

algorithmic efficiency for miners and merchants

Summary

the longest-chain rule cannot scale,

but Bitcoin can

chains are neat and compact, yet vulnerable

using DAGs allows utilization of the entire

mining power

“Bitcoiners of the world, unite! You have nothing to lose but your chains!“ (~~Karl Marx~~)

Thank You